Amendments to the Claims:

1. (Currently Amended) A method comprising:

obtaining performance data for software that has executed in a data processing system, wherein the performance data comprises a plurality of instruction addresses and corresponding performance information for each instruction address of the plurality of instruction addresses;

obtaining dump information from the data processing system, wherein the dump information was produced during execution of the software, and wherein the dump information comprises ~~the~~ a plurality of instructions, with each instruction of the plurality of instructions having a ~~and~~ corresponding instruction address of the plurality of instruction addresses;

automatically identifying common code segments in the dump information, wherein a common code segment comprises an ordered set of multiple instructions that appears multiple times in the dump information; and

generating aggregate performance data for the common code segments, based at least in part on the instruction addresses associated with the common code segments from the dump information, the instruction addresses from the performance data, and the corresponding performance information from the performance data.

2. (Original) A method according to claim 1, wherein:

the operation of obtaining performance data comprises obtaining performance data for instructions generated by a dynamic compiler; and

the operation of generating aggregate performance data for the common code segments comprises generating aggregate performance data for common code segment generated by the dynamic compiler.

3. (Original) A method according to claim 1, wherein the operation of identifying common code segments in the dump information comprises:

selecting a candidate code segment from the dump information;

determining whether the candidate code segment occurs multiple times in the dump information; and

identifying the candidate code segment as a common code segment in response to determining that the candidate code segment occurs multiple times in the dump information.

4. (Original) A method according to claim 1, wherein the operation of identifying common code segments in the dump information comprises:

selecting a candidate code segment from the dump information;

determining whether the dump information includes at least one additional absolute match for the candidate code segment; and

identifying the candidate code segment as a common code segment in response to determining that the dump information includes at least one additional absolute match for the candidate code segment.

5. (Original) A method according to claim 1, wherein the operation of identifying common code segments in the dump information comprises:

selecting a candidate code segment from the dump information;

identifying elements in the candidate code segment as significant;

determining whether the dump information includes at least one additional match for the candidate code segment, wherein the additional match comprises instructions with elements matching the significant elements in the candidate code segment; and

identifying the candidate code segment as a common code segment in response to determining that the dump information includes at least one additional match for the candidate code segment.

6. (Original) A method according to claim 1, wherein the performance information comprises one or more measurements selected from the group consisting of:

execution time data for individual instructions; and

cache miss data for individual instructions.

7. (Original) A method according to claim 1, wherein:

the operation of identifying common code segments in the dump information comprises identifying at least first and second common code segments; and

the operation of generating aggregate performance data for the common code segments comprises:

collecting performance data for multiple instances of the first common code segment;

generating aggregate performance data for the first common code segment, based at least in part on the performance data for the multiple instances of the first common code segment;

collecting performance data for multiple instances of the second common code segment; and

generating aggregate performance data for the second common code segment, based at least in part on the performance data for the multiple instances of the second common code segment.

8. (Currently Amended) A method according to claim 7, wherein the operation of generating aggregate performance data for the common code segments comprises:

collecting performance information corresponding to instruction addresses for ~~substantially all~~ a plurality of instances of the common code segment in the dump information.

9. (Currently Amended) An apparatus, comprising:

a machine accessible storage medium; and

software encoded in the machine accessible storage medium, wherein the software, when executed by a processing system, performs operations comprising:

obtaining performance data for software that has executed in a data processing system, wherein the performance data comprises a plurality of instruction addresses and corresponding performance information for each instruction address of the plurality of instruction addresses;

obtaining dump information from the data processing system, wherein the dump information was produced during execution of the software, and wherein the dump information comprises ~~the~~ a plurality of instructions, with each instruction of the plurality of instructions having a ~~and~~ corresponding instruction address of the plurality of instruction addresses;

automatically identifying common code segments in the dump information, wherein a common code segment comprises an ordered set of multiple instructions that appears multiple times in the dump information; and

generating aggregate performance data for the common code segments, based at least in part on the instruction addresses associated with the common code segments from the dump information, the instruction addresses from the performance data, and the corresponding performance information from the performance data.

10. (Original) An apparatus according to claim 9, wherein:

the operation of obtaining performance data comprises obtaining performance data for instructions generated by a dynamic compiler; and

the operation of generating aggregate performance data for the common code segments comprises generating aggregate performance data for common code segment generated by the dynamic compiler.

11. (Original) An apparatus according to claim 9, wherein the operation of identifying common code segments in the dump information comprises:

selecting a candidate code segment from the dump information;

determining whether the candidate code segment occurs multiple times in the dump information; and

identifying the candidate code segment as a common code segment in response to determining that the candidate code segment occurs multiple times in the dump information.

12. (Original) An apparatus according to claim 9, wherein the operation of identifying common code segments in the dump information comprises:

selecting a candidate code segment from the dump information;

determining whether the dump information includes at least one additional absolute match for the candidate code segment; and

identifying the candidate code segment as a common code segment in response to determining that the dump information includes at least one additional absolute match for the candidate code segment.

13. (Original) An apparatus according to claim 9, wherein the operation of identifying common code segments in the dump information comprises:

selecting a candidate code segment from the dump information;

identifying elements in the candidate code segment as significant;

determining whether the dump information includes at least one additional match for the candidate code segment, wherein the additional match comprises instructions with elements matching the significant elements in the candidate code segment; and

identifying the candidate code segment as a common code segment in response to determining that the dump information includes at least one additional match for the candidate code segment.

14. (Original) An apparatus according to claim 9, wherein the performance information comprises one or more measurements selected from the group consisting of:

execution time data for individual instructions; and

cache miss data for individual instructions.

15. (Original) An apparatus according to claim 9, wherein:

the operation of identifying common code segments in the dump information comprises identifying at least first and second common code segments; and

the operation of generating aggregate performance data for the common code segments comprises:

collecting performance data for multiple instances of the first common code segment;

generating aggregate performance data for the first common code segment, based at least in part on the performance data for the multiple instances of the first common code segment;

collecting performance data for multiple instances of the second common code segment; and

generating aggregate performance data for the second common code segment, based at least in part on the performance data for the multiple instances of the second common code segment.

16. (Currently Amended) An apparatus according to claim 15, wherein the operation of generating aggregate performance data for the common code segments comprises:

collecting performance information corresponding to instruction addresses for ~~substantially all~~ a plurality of instances of the common code segment in the dump information.

17. (Currently Amended) A system, comprising:

a processor;

a machine accessible medium responsive to the processor; and

instructions in the machine accessible medium, wherein the instructions, when executed by the processor, perform operations comprising:

obtaining performance data for software that has executed in a data processing system, wherein the performance data comprises a plurality of instruction addresses and corresponding performance information for each instruction address of the plurality of instruction addresses;

obtaining dump information from the data processing system, wherein the dump information was produced during execution of the software, and wherein the dump information comprises ~~the~~ a plurality of instructions, with each instruction of the plurality of instructions having a ~~and~~ corresponding instruction address of the plurality of instruction addresses;

automatically identifying common code segments in the dump information, wherein a common code segment comprises an ordered set of multiple instructions that appears multiple times in the dump information; and generating aggregate performance data for the common code segments, based at least in part on the instruction addresses associated with the common code segments from the dump information, the instruction addresses from the performance data, and the corresponding performance information from the performance data.

18. (Original) A system according to claim 17, wherein:

the operation of obtaining performance data comprises obtaining performance data for instructions generated by a dynamic compiler; and

the operation of generating aggregate performance data for the common code segments comprises generating aggregate performance data for common code segment generated by the dynamic compiler.

19. (Original) A system according to claim 17, wherein the operation of identifying common code segments in the dump information comprises:

selecting a candidate code segment from the dump information;

determining whether the candidate code segment occurs multiple times in the dump information; and

identifying the candidate code segment as a common code segment in response to determining that the candidate code segment occurs multiple times in the dump information.

20. (Original) A system according to claim 17, wherein:

the operation of identifying common code segments in the dump information comprises identifying at least first and second common code segments; and

the operation of generating aggregate performance data for the common code segments comprises:

collecting performance data for multiple instances of the first common code segment;

generating aggregate performance data for the first common code segment, based at least in part on the performance data for the multiple instances of the first common code segment;

collecting performance data for multiple instances of the second common code segment; and generating aggregate performance data for the second common code segment, based at least in part on the performance data for the multiple instances of the second common code segment.